Q1 [Find the "Kth" max and min element of an array](https://practice.geeksforgeeks.org/problems/kth-smallest-element/0)

class Solution{

public:

vector<int> kLargest(int arr[], int n, int k) {

// code here

vector<int>ans;

sort(arr,arr+n);

for(int i = n-1; i >= n-k; i--) {

ans.push\_back(arr[i]);

}

return ans;

}

};

A screenshot of a computer

Description automatically generated

Q2 [Given an array which consists of only 0, 1 and 2. Sort the array without using any sorting algo](https://practice.geeksforgeeks.org/problems/sort-an-array-of-0s-1s-and-2s/0)

class Solution {

public:

void sortColors(vector<int>& nums) {

int low = 0, mid = 0;

int si = nums.size();

int high = si - 1;

while (mid <= high) {

if (nums[mid] == 0) {

swap(nums[low], nums[mid]);

low++;

mid++;

} else if (nums[mid] == 1) {

mid++;

} else {

swap(nums[mid], nums[high]);

high--;

}

}

}

};

A screenshot of a computer

Description automatically generated

Q3 [find Largest sum contiguous Subarray [V. IMP]](https://practice.geeksforgeeks.org/problems/kadanes-algorithm/0)

class Solution {

public:

int maxSubArray(vector<int>& nums) {

int maxSum = INT\_MIN;

int currentSum = 0;

for (int i = 0; i < nums.size(); i++) {

currentSum += nums[i];

if (currentSum > maxSum) {

maxSum = currentSum;

}

if (currentSum < 0) {

currentSum = 0;

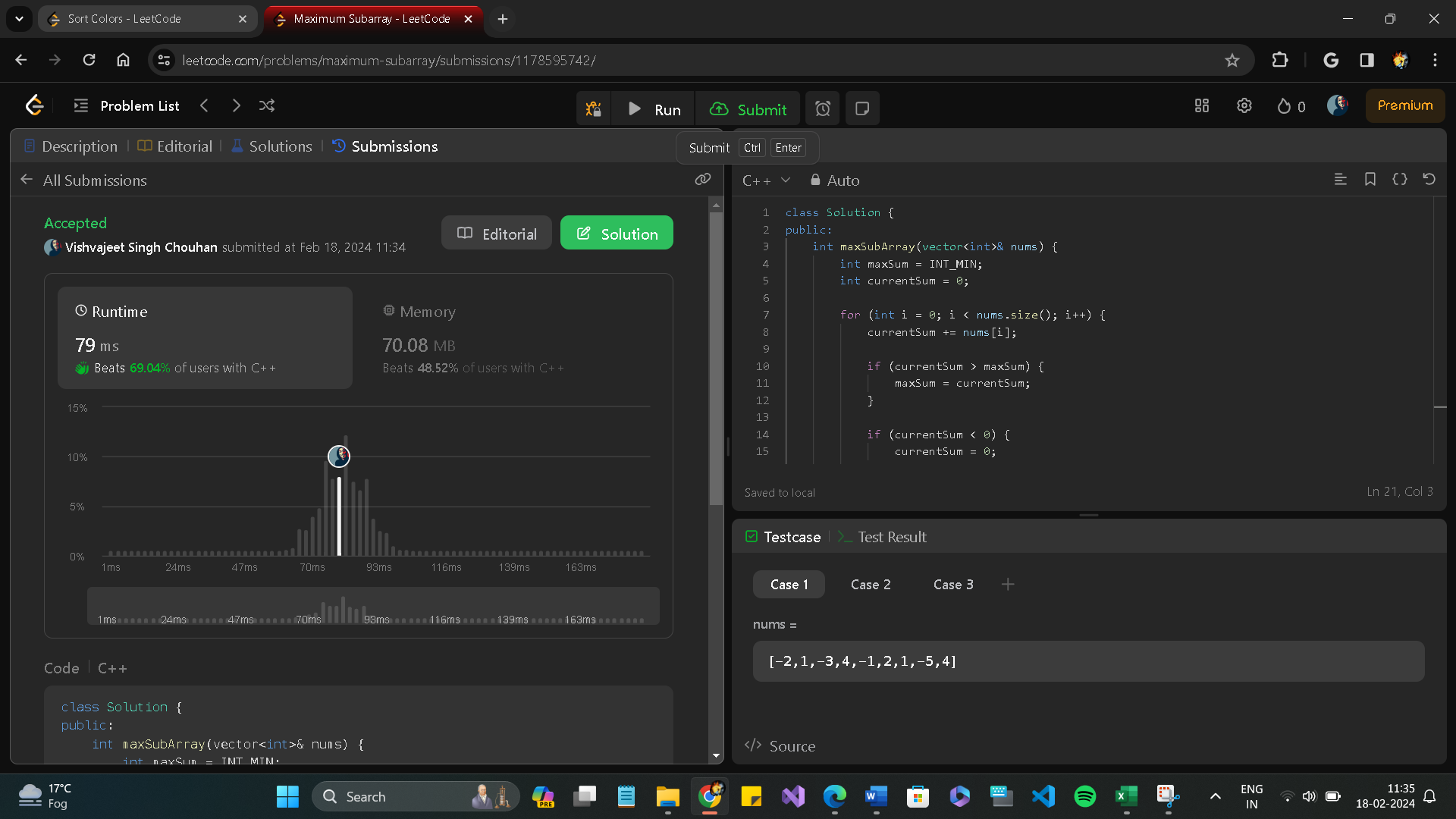
}

}

return maxSum;

}

};



Q4 [Count Inversion](https://practice.geeksforgeeks.org/problems/inversion-of-array/0)

class Solution {

public:

bool isIdealPermutation(vector<int>& A) {

int loc = 0;

for (int i = 0; i < A.size() - 1; ++i) {

if (A[i] > A[i + 1]) {

loc++;

}

}

int glo = f(A, 0, A.size() - 1);

return loc == glo;

}

// count inversions while merge sorting

int f(vector<int>& nums, int i, int j) {

if (i >= j) {

return 0;

}

int med = (i + j) >> 1;

long long count = f(nums, i, med) + f(nums, med + 1, j);

int ii = i;

for (int k = med + 1; k <= j; ++k) {

while (ii <= med && nums[ii] < nums[k]) {

ii++;

}

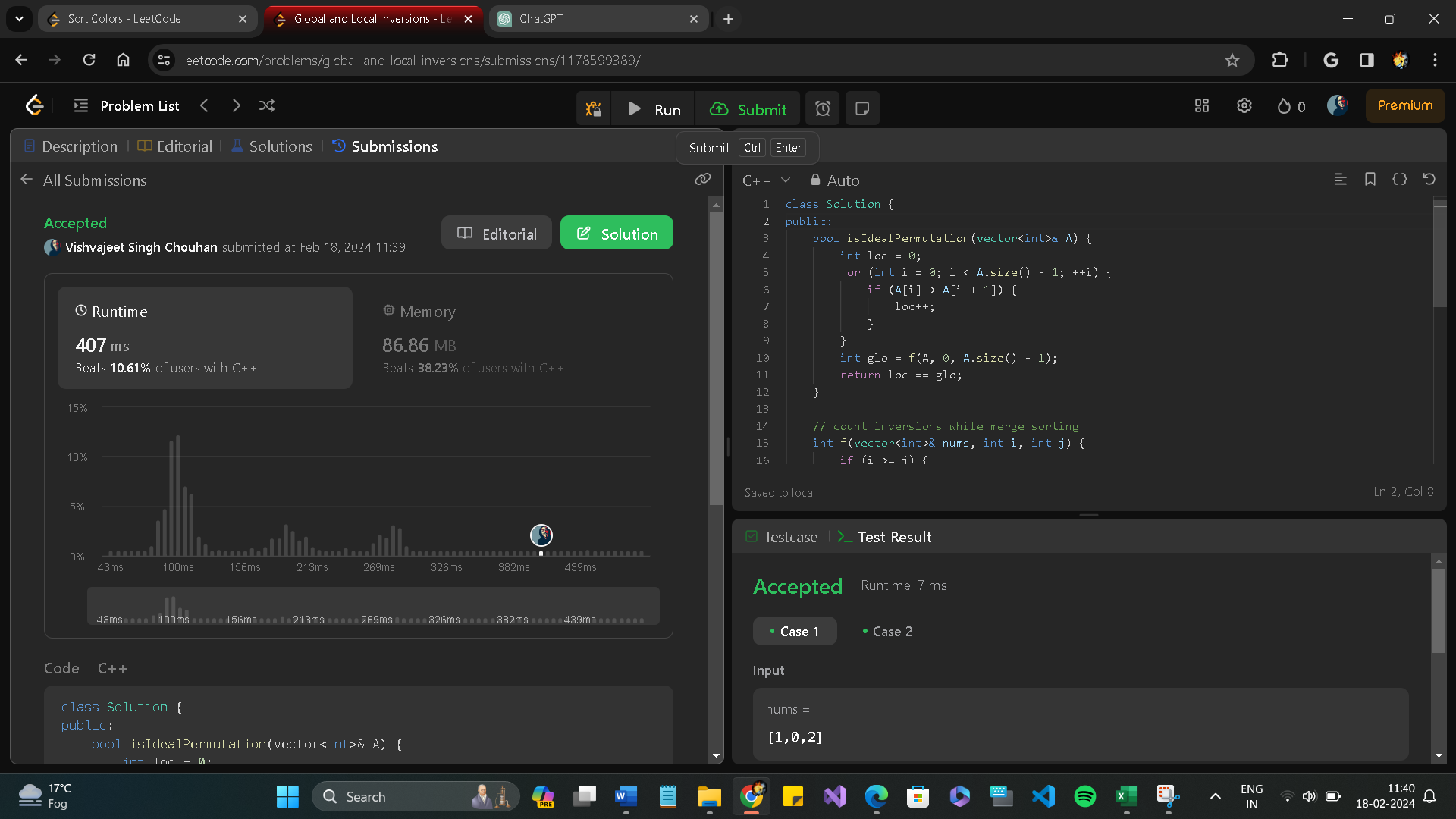
count += (med - ii + 1);

}

sort(nums.begin() + i, nums.begin() + j + 1);

return count;

}



Q5 [find all pairs on integer array whose sum is equal to given number](https://practice.geeksforgeeks.org/problems/count-pairs-with-given-sum5022/1)

class FindSumPairs {

public:

vector<int> temp1, temp2;

map<int, int> mp;

FindSumPairs(vector<int>& nums1, vector<int>& nums2) {

temp1 = nums1;

temp2 = nums2;

for(auto i:nums2){

mp[i]++;

}

}

void add(int index, int val) {

mp[temp2[index]]--;

temp2[index] += val;

mp[temp2[index]]++;

}

int count(int tot) {

int cnt = 0;

for(int i=0; i<temp1.size(); ++i){

int k = tot - temp1[i];

auto it = mp.find(k);

if(it != mp.end()){

cnt += it->second;

}

}

return cnt;

}};A screenshot of a computer

Description automatically generated

Q6[find maximum product subarray](https://practice.geeksforgeeks.org/problems/maximum-product-subarray3604/1)

class Solution {

public:

int maxProduct(vector<int>& nums) {

int maxi = INT\_MIN;

int prod=1;

for(int i=0;i<nums.size();i++)

{

prod\*=nums[i];

maxi=max(prod,maxi);

if(prod==0)

prod=1;

}

prod=1;

for(int i=nums.size()-1;i>=0;i--)

{

prod\*=nums[i];

maxi=max(prod,maxi);

if(prod==0)

prod=1;

}

return maxi};A screenshot of a computer

Description automatically generated

Q7 [Given an array of size n and a number k, fin all elements that appear more than " n/k " times.](https://www.geeksforgeeks.org/given-an-array-of-of-size-n-finds-all-the-elements-that-appear-more-than-nk-times/)

int countOccurence(int arr[], int n, int k) {

// Your code here

unordered\_map<int,int>mp;

for(int i=0;i<n;i++)

{

mp[arr[i]]++;

}

int count=0;

for(auto it=mp.begin();it!=mp.end();it++)

{

int val=it->second;

if(val>n/k)

count++;

} return count}};

A screenshot of a computer

Description automatically generated

Q6 [Find the triplet that sum to a given value](https://practice.geeksforgeeks.org/problems/triplet-sum-in-array/0)

class Solution {

public:

//Function to find if there exists a triplet in the

//array A[] which sums up to X.

bool find3Numbers(int A[], int n, int X)

{

//Your Code Here

for(int i=n-2 ;i>=0 ;i--)

{

for(int j = 0; j<=i; j++)

{

if(A[j]>A[j+1])

swap(A[j],A[j+1]);

}

}

for(int i =0 ;i<n-2;i++) //here there is sum of three number so here we use n-2

{

int ans =X-A[i];

int start = i+1 ,end = n-1;

while(start<end)

{

if(A[start]+A[end]==ans)

return 1;

else if(A[start]+A[end]>ans)

end--;

else

start++;

}

}

return 0;

}

};

A screenshot of a computer

Description automatically generated

Q7 [Smallest Subarray with sum greater than a given value](https://practice.geeksforgeeks.org/problems/smallest-subarray-with-sum-greater-than-x/0)

int smallestSubWithSum(int arr[], int n, int x)

{

if(n==1){

if(arr[0] > x){

return 1;

}

else{

return 0;

}

}

// Check if whole array sum < x ?

int wholeSum = 0;

for(int i=0; i<n; i++){

wholeSum += arr[i];

}

if(wholeSum < x){

return 0;

}

int i = 0;

int j = 0;

int sum = 0;

int mini = INT\_MAX;

while(i<n && j<n){

sum = sum + arr[j];

if(sum > x){

mini = min(mini,j-i+1);

sum = 0;

i = i+1;

j=i;

}

else{

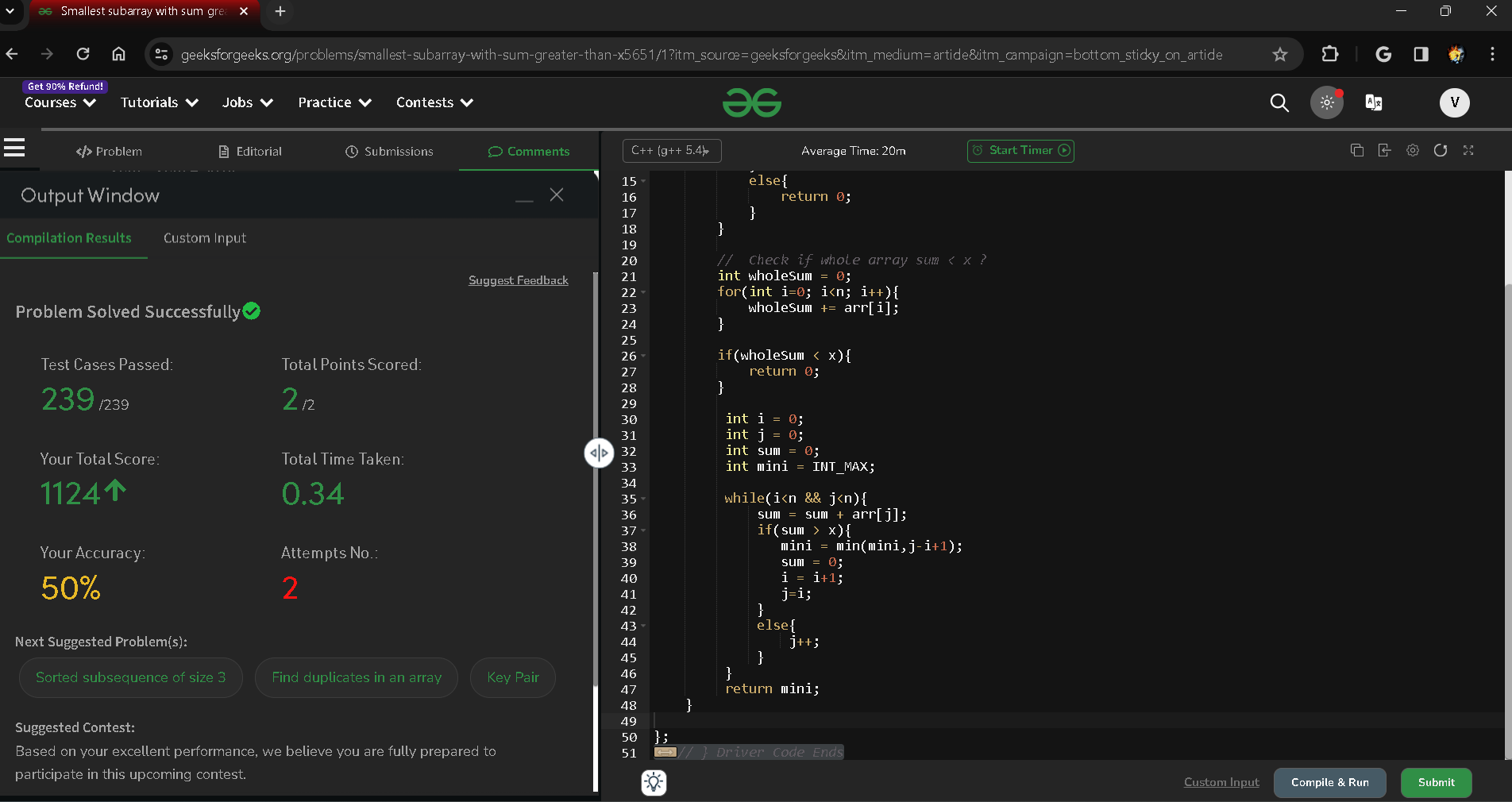
j++;

}

}

return mini;

}



Q8 [Write a Code to check whether one string is a rotation of another](https://www.geeksforgeeks.org/a-program-to-check-if-strings-are-rotations-of-each-other/)

bool checkString(string& s1, string& s2, int indexFound,int Size)

{

for (int i = 0; i < Size; i++) {

// check whether the character is equal or not

if (s1[i] != s2[(indexFound + i) % Size])

return false;

// %Size keeps (indexFound+i) in bounds, since it

// ensures it's value is always less than Size

}

return true;

}

Q9 [Write a Program to check whether a string is a valid shuffle of two strings or not](https://www.programiz.com/java-programming/examples/check-valid-shuffle-of-strings)

bool isShuffledSubstring(string A, string B)

{

int n = A.length();

int m = B.length();

// Return false if length of

// string A is greater than

// length of string B

if (n > m) {

return false;

}

else {

// Sort string A

sort(A.begin(), A.end());

// Traverse string B

for (int i = 0; i < m; i++) {

// Return false if (i+n-1 >= m)

// doesn't satisfy

if (i + n - 1 >= m)

return false;

// Initialise the new string

string str = "";

// Copy the characters of

// string B in str till

// length n

for (int j = 0; j < n; j++)

str.push\_back(B[i + j]);

// Sort the string str

sort(str.begin(), str.end());

// Return true if sorted

// string of "str" & sorted

// string of "A" are equal

if (str == A)

return true;

}

}

}

Q10 [Balanced Parenthesis problem.[Imp]](https://practice.geeksforgeeks.org/problems/parenthesis-checker/0)

class Solution {

public:

bool isValid(string s) {

stack<char>helper;

int i=0;

while(i<s.length()){

if(s[i]==')' && helper.size()>0 && helper.top()=='('){

helper.pop();

}

else if(s[i]=='}' && helper.size()>0 && helper.top()=='{'){

helper.pop();

}

else if(s[i]==']' && helper.size()>0 && helper.top()=='['){

helper.pop();

}

else{

helper.push(s[i]);

}

i++;

}

if(helper.size()==0){

return true;

}

return false;

}

};

A screenshot of a computer

Description automatically generated

Q11[KMP Algo](https://practice.geeksforgeeks.org/problems/longest-prefix-suffix2527/1)

class Solution {

public:

string longestPrefix(string s) {

vector<int> pi(s.length() + 1, 0);

pi[0] = -1;

int k = -1;

for (int i = 1; i <= s.length(); i++) {

while(k >= 0 && s[k] != s[i - 1]) k = pi[k];

pi[i] = ++k;

}

return s.substr(0, pi[s.length()]);

}

};

A screenshot of a computer

Description automatically generated

Q12 [Longest Common Prefix](https://leetcode.com/problems/longest-common-prefix/)

class Solution {

public:

string longestCommonPrefix(vector<string>& strs) {

std::string s = "";

for (int i = 0; i < strs[0].length(); i++) {

for (int j = 1; j < strs.size(); j++) {

if (i >= strs[j].length() || strs[j][i] != strs[0][i]) {

return s;

}

}

s += strs[0][i]; }

return s;

}

};

A screenshot of a computer

Description automatically generated

Q13 [Rearrange characters in a string such that no two adjacent are same](https://practice.geeksforgeeks.org/problems/rearrange-characters/0)

string rearrangeString(string str)

{

//code here

//code here

int n=str.length();

priority\_queue<pair<int,int>> pq;

vector<int> count(26,0);

for(int i=0;i<n;i++){

count[str[i]-'a']++;

}

for(int i=0;i<26;i++){

if(count[i]>0){

pq.push({count[i],i});

}

}

int index=0;

while(!pq.empty()){

int cnt=pq.top().first;

int ch=pq.top().second;

pq.pop();

while(cnt>0){

if(index>=n){

index=1;

}

str[index]=ch+'a';

cnt--,index+=2;

}

}

for (int i=0;i<n-1;i++){

if(str[i]==str[i+1]){

return "-1";

}

}

return str;

}

A screenshot of a computer

Description automatically generated

Q 14 [Find the smallest window in a string containing all characters of another string](https://practice.geeksforgeeks.org/problems/smallest-window-in-a-string-containing-all-the-characters-of-another-string/0)

string minWindow(string s, string t) {

if (s.empty() || t.empty() || s.length() < t.length()) {

return "";

}

std::vector<int> map(128, 0);

int count = t.length();

int start = 0, end = 0, minLen = INT\_MAX, startIndex = 0;

/// UPVOTE !

for (char c : t) {

map[c]++;

}

while (end < s.length()) {

if (map[s[end++]]-- > 0) {

count--;

}

while (count == 0) {

if (end - start < minLen) {

startIndex = start;

minLen = end - start;

}

if (map[s[start++]]++ == 0) {

count++;

}

}

}

return minLen == INT\_MAX ? "" : s.substr(startIndex, minLen);

}

